Analyse et prédiction de l'impact de changements dans un système à objets : Approche probabiliste

M.K Abdi*, H. Lounis**, H. Sahraoui*

* DIRO, Université de Montréal,
CP 6128, succ. Centre-Ville, Montréal QC H3C 3J7, Canada
{abdimust, sahraouh} @iro.umontreal.ca

** Département d'Informatique, Université du Québec à Montréal
CP 8888, succ. Centre-ville, Montréal QC H3C 3P, Canada
lounis.hakim@uqam.ca

Résumé. Nous proposons dans cet article une approche probabiliste utilisant les réseaux bayésiens pour analyser et prédire les impacts des changements dans les systèmes à objets. Un modèle d'impact a été construit et des probabilités ont été affectées aux différents sommets du réseau. Des données récoltées sur un système réel sont utilisées pour étudier empiriquement des hypothèses (relations) de causalité entre d'une part, des attributs internes de logiciel, et d'autre part, l'impact du changement. Pour ce faire, plusieurs scénarios ont été exécutés sur le réseau. Les résultats obtenus ont d'une part, confirmé certains résultats déjà trouvés lors de travaux antérieurs, mais d'autre part, remis en cause d'autres conclusions. Cette étude entre dans le cadre général d'une démarche tendant à proposer des modèles d'estimation de la qualité du produit logiciel ; elle montre qu'un modèle probabiliste constitue une alternative intéressante, aux modèles non probabilistes proposés dans la littérature.

1 Introduction

La maintenance est la dernière phase du cycle de vie d'un logiciel. Elle est définie comme étant le processus de modification d'un logiciel en exploitation pour lui permettre de satisfaire les spécifications actuelles et futures (Computer, 1993). Selon Pfleeger (Pfleeger, 1990), le coût de la maintenance dépend du degré de dépendance entre les entités d'une architecture logicielle. La modification des systèmes est une tâche à la fois difficile et porteuse de conséquence sur la suite de l'évolution de ces systèmes (Wilde et Huitt, 1992). Les effets des changements subis par le système doivent donc être pris en considération. En effet, un petit changement peut avoir des effets considérables et inattendus sur le reste des éléments du système. Le danger encouru lors de la modification réside dans cette conséquence de l'impact d'un changement donné. La modularité en conception objet, adéquatement utilisée, limite les effets relatifs aux changements. Néanmoins, en général, ces effets sont subtils et difficiles à découvrir. Pour toutes ces raisons, les concepteurs ont besoin

de mécanismes pour analyser les changements et connaître leurs impacts sur le reste du système.

Nous définissons un changement dans un programme comme une modification apportée à un de ses éléments (classe, méthode ou variable). Comme exemples de changement, on peut avoir la suppression d'une variable, la modification de la portée d'une méthode, de « public » à « protected » ou le déplacement du lien entre une classe et son parent. L'impact est vu dans notre contexte comme la conséquence d'un changement. L'analyse de l'impact est une activité dont l'objectif est de déterminer l'étendue d'une requête de changement. Elle estime les éléments affectés, au niveau du code source et de la documentation, lorsqu'un changement est effectué. Nous signalons enfin qu'un modèle d'impact permet de prédire quelles classes seraient affectées si un changement avait été réellement fait.

Nous rappelons aussi que le couplage est défini comme le degré d'interdépendance entre les composants (classes) du système. Plus une classe est couplée avec d'autres classes, plus elle est sensible aux changements effectués dans ces classes et plus elle est susceptible de subir des erreurs. Par ailleurs, une telle classe est plus difficile à comprendre, à modifier et à déboguer et par conséquent difficile à maintenir.

Notre travail entre dans le contexte du standard ISO/IEC (14598), qui énonce que les mesures d'attributs internes d'un produit logiciel, comme le couplage, la cohésion ou l'héritage, ne sont utiles que si elles sont reliées à des attributs externes de qualité, comme la maintenabilité ou la réutilisabilité, par exemple. La motivation de notre travail est ainsi, d'améliorer la maintenance des systèmes à objets et d'intervenir plus précisément dans la tâche de l'analyse et de la prédiction de l'impact du changement. En identifiant l'impact potentiel d'une modification, on réduit le risque d'entamer des changements coûteux et imprévisibles. Pour cela, nous essayons de donner plus d'explications sur les facteurs réels et responsables de cet impact du changement ainsi que son évolution. Il existe plusieurs modèles de représentation et les réseaux bayésiens (RBs) constituent une approche quantitative particulière qui peut intégrer l'incertitude dans le raisonnement (Naïm et al., 2004), offrant ainsi des explications proches de la réalité. De plus, avec les RBs, il est aussi possible d'exploiter les jugements des experts pour anticiper les prédictions, qui dans notre cas portent sur l'impact du changement. Par ailleurs, les RBs ont la capacité d'apprentissage incrémental à partir de données. Ceci est vrai aussi bien pour l'apprentissage des paramètres que pour l'apprentissage de la structure, ce qui facilite l'évolution du modèle. Cette capacité contribuera à l'amélioration, en fonction du temps, de la structure du réseau bayésien et de ses paramètres, et ce, par l'acquisition de nouvelles données. Dans le présent travail, nous nous focalisons beaucoup plus sur la conception du modèle probabiliste (RB) et sur sa mise en œuvre sur un système réel. L'aspect apprentissage n'est pas traité dans cette étude.

La section 2 fait un tour d'horizon de différents travaux consacrés à l'analyse de l'impact du changement. Notre proposition fait l'objet de la troisième section. Nous commençons par présenter les principales étapes de notre approche, suivies d'un bref rappel sur les RBs. Puis nous montrons comment construire le graphe (RB) dans le cadre de notre expérimentation. Ensuite, nous expliquons l'affectation des paramètres (probabilités) aux différents sommets du réseau. La section 4 est réservée à la mise en œuvre du réseau et à la discussion des résultats obtenus. Les perspectives de notre travail sont discutées en conclusion.

2 Travaux connexes

Plusieurs travaux ont été menés concernant l'impact du changement. Ainsi, Han (Han, 1997) a proposé une approche pour calculer l'impact du changement à partir des documents de conception et d'implantation. Cette approche considère la représentation originale des artéfacts logiciels (classes) plutôt qu'un modèle de système extrait séparément. Les dépendances d'artéfacts impliquent l'héritage, l'agrégation et l'association. L'approche est à la fois automatique, utilisant des règles de propagation de changement, et manuelle, exigeant l'intervention de l'utilisateur. Cependant, aucun modèle de changement n'est défini formellement. D'un autre côté, Lindvall (Lindvall, 1999) a identifié les changements les plus communs et fréquents en C++, afin que les modèles de changement puissent être conçus pour aider les développeurs à prévoir les futurs besoins.

Dans (Antoniol et al., 1999), les auteurs ont prédit l'évolution dans le temps de la taille des systèmes à objets, à partir de l'analyse des classes impactées par une demande de changement. Ils ont prédit la taille des changements en termes de lignes de code ajoutées/modifiées. Kung et al., intéressés par les tests de régression, ont développé un modèle d'impact de changements tenant compte de trois liens : héritage, association et agrégation (Kung et al., 1995). Lee et Offutt (Lee et Offutt, 1996; Lee, 1998), pour examiner les effets de l'encapsulation, de l'héritage et du polymorphisme sur l'impact de changements, ont proposé des algorithmes pour calculer l'impact complet de changements faits dans une classe donnée.

Dans (Cantave, 2001), l'analyse de l'impact a été faite dans l'objectif de réduire les coûts et la durée des tests de régression. L'étude a été faite à partir d'un graphe de dépendance. Briand et al. dans (Brian et al., 1999), ont essayé de voir si les mesures de couplage capturant toutes sortes de collaborations entre classes, peuvent aider à faire l'analyse d'impact du changement. Cette étude a montré qu'un certain nombre de métriques de couplage liées à l'agrégation, mais aussi au couplage d'invocation, sont reliées à la propagation de modification. Cela permet de localiser l'analyse de dépendance et de réduire l'effort d'analyse d'impact.

Dans (Chaumun et al., 1999; Kabaili et al., 2000; Kabaili, 2002), un modèle de changements et d'impact de changements, a été défini au niveau conceptuel pour étudier la changeabilité des systèmes à objets. L'approche adoptée utilise des propriétés caractéristiques de la conception des systèmes à objets (complexité, cohésion, couplage, etc.) mesurées par des métriques afin de prédire la changeabilité. D'après la norme ISO9126 (ISO9126, 2001), la changeabilité est définie comme la facilité avec laquelle un logiciel peut être modifié pour répondre à un besoin donné, incluant l'ajout de fonctionnalité, la correction d'erreurs et l'adaptation à un nouvel environnement. Selon une perspective différente, Sahraoui et al. ont étudié dans (Sahraoui et al., 2000), l'impact de la restructuration de parties du code sur les métriques structurelles. Cette étude a permis de déterminer quelles sont les restructurations qui peuvent améliorer ou détériorer certaines propriétés structurelles. Récemment, dans (Abdi et al., 2006a; 2006b; 2007), les auteurs ont montré aussi que certains types de couplage (couplage d'importation méthode-méthode, couplage d'invocation, etc.) influencent l'impact du changement.

D'autre part, dans (Fenton et Neil, 2000a), Fenton et Neil montrent bien les avantages de l'approche modèle-causalité utilisant les RBs par rapport à l'approche classique (naïve) s'appuyant sur la régression. Dans d'autres travaux (Fenton et Neil, 2000b; 2001; Neil et al., 2000), Fenton et son équipe montrent à travers des études de cas que les réseaux bayésiens

peuvent fournir des prédictions appropriées en présence d'incertitude, de jugements subjectifs d'experts et d'incomplétude de l'information qui sont inhérentes au génie logiciel.
Dans le présent travail, nous voulons explorer cette voie de recherche et montrer ainsi les avantages de l'approche probabiliste par rapport à l'approche adoptée dans nos travaux antérieurs (Abdi et al., 2006a; 2006b; 2007). Plus précisément, nous essayons à partir des mesures faites sur une propriété architecturale (couplage) de prédire l'impact de changements suite à la mise en œuvre du RB. Dans la section suivante, nous présentons notre approche (proposition) en expliquant, pas à pas, ses différentes étapes.

3 Proposition

Les principales étapes sur lesquelles repose notre approche sont les suivantes :

1- Construction de la structure du graphe (RB) à partir de connaissances pratiques (études empiriques)
2- Affectation des paramètres (tables de probabilité de sommet, logique floue)
3- Inférence bayésienne (algorithmes, outils)
4- Résultats.

Notons que le retour depuis l'étape 4 à l'étape 1 est tout à fait possible dans un but de remise en cause ou tout simplement pour faire d'autres exécutions du réseau. Les trois premières étapes sont expliquées dans la présente section. Par contre, les deux dernières sont présentées dans la section 4. Afin de faciliter la compréhension des notions abordées dans notre approche, un rappel des concepts des réseaux bayésiens est à notre avis indispensable ; c'est l'objet de la section qui suit.

3.1 Rappel sur les réseaux bayésiens

Les RBs sont basés sur le théorème de Bayes. Ce théorème décrit les relations qui existent entre les probabilités simples et conditionnelles. Si A et B sont deux évènements et si l'on connaît les probabilités de A, de B et de B sachant A, le théorème de Bayes permet de déterminer la probabilité de A sachant B :

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}$$

Les RBs sont le résultat du mariage entre la théorie des graphes et la théorie des probabilités (Naïm et al., 2004). Un RB est un graphe causal où :

- Les sommets représentent des variables aléatoires. Une variable aléatoire a un certain nombre d'états, par exemple « Oui » et « Non », et une probabilité de distribution pour ces états où la somme des probabilités de tous les états doit être égale à 1. De cette façon, un modèle de RB est sujet aux axiomes standards de la théorie des probabilités.
- Les arcs définissent des relations causales entre les sommets. On parle de « parents » et de « fils » quand on se réfère à des liens. Un arc va du parent vers le fils. Les sommets parents qui affectent un même sommet fils doivent être des variables indépendantes en termes de

leur action sur ce fils. Rappelons qu'en théorie des probabilités, deux évènements A et B sont indépendants si la réalisation de A n'affecte aucunement la probabilité de B.

- Chaque sommet est lié à une Table de Probabilité de Sommet (TPS), qui modélise la relation incertaine entre le sommet et ses parents. L'incertitude dans la relation vient du fait que nous ne pouvons déterminer la valeur d'un sommet fils qu'avec une certaine probabilité, sachant les valeurs des sommets parents, avec également une certaine probabilité. Les tables de probabilité conditionnelle liées aux sommets d'un RB déterminent la force des liens du graphe et sont utilisées pour calculer la probabilité de distribution de chaque sommet dans le RB. Ceci est réalisé en spécifiant la probabilité conditionnelle d'un sommet sachant tous ses parents : $P(X \mid A, B)$, X étant le sommet fils de A et B. Si un sommet n'a aucun parent, une table antérieure de probabilité 'a priori' de ce sommet lui sera associée. La section 3.3 présente ces notions avec plus de détails. L'intérêt particulier des RBs est de tenir compte simultanément des connaissances a priori d'experts (dans le graphe ou la structure) et de l'expérience contenue dans les données (les paramètres).

3.2 Construction du graphe (RB)

Généralement, la construction d'un RB se fait en deux étapes : produire la structure du graphe approprié, puis affecter les valeurs des probabilités aux sommets du réseau (Naïm et al., 2004). L'affectation de ces valeurs se fait selon les experts du domaine ou à partir d'études empiriques. À ce niveau, il est important de vérifier que les sommets parents qui affectent un même sommet fils sont des variables indépendantes au sens des probabilités. De plus, afin de respecter le formalisme de construction des RBs, lors de la construction des liens entre les sommets, il faut veiller à vérifier l'absence de circuits entre les sommets du réseau.

Comme cela est déjà signalé (en section 2), nous avons vérifié dans (Abdi et al., 2006a; 2006b; 2007), l'hypothèse prétendant que *le couplage influence l'impact du changement dans un système à objets*. Nous rappelons à ce niveau que comme les liens interclasses sont censés être plus responsables de la propagation de la modification que les liens intra-classes, nous nous sommes concentrés dans nos travaux antérieurs sur la propriété de couplage, et nous avons essayé de voir s'il y a des relations de cause à effet entre cette propriété architecturale du système et l'impact du changement. Cependant, si nous considérons à la fois toutes les métriques mesurant les différentes facettes du couplage entre classes, la construction du RB risque d'être laborieuse et la structure du graphe complexe. D'autre part, les résultats affirment que parmi l'ensemble des dix types de couplage considérés, mesurés respectivement par les métriques définies dans la table 1, cinq types influencent effectivement l'impact du changement. Certaines de ces métriques sont considérées comme des métriques de conception (AMMIC et OMMIC), car pouvant être utilisées dès la phase de conception, alors que d'autres, sont des métriques d'implémentation (MPC, CBOU, CBONA), car nécessitant la présence du code source. La figure 1 présente le graphe exprimant cette connaissance sous forme de RB.

Pour être conformes à l'hypothèse de Bayes, nous considérons que les métriques mesurent différents types d'aspects de couplage et que leurs valeurs sont donc indépendantes les unes des autres. Par ailleurs, il est à noter que la relation définitionnelle/causale entre les sommets d'un RB est de nature incertaine. Rappelons que dans un RB, le lien entre deux sommets véhicule de l'incertitude exprimée par des probabilités. La structure du graphe est

donc complétée par la suite par des fonctions probabilistes. De telles fonctions permettent de déterminer le degré avec lequel un sommet parent influence ou définit un sommet fils. Dans la figure ci-dessous, la relation entre les sommets parents et fils est causale (cas du sommet *Impact*) ou définitionnelle (cas des sommets *DesignMetrics* et *ImplementationMetrics*).

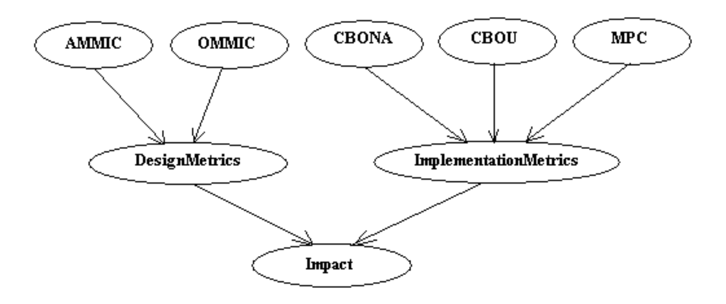


FIG. 1 - *Réseau d'impact du changement.*

Métriques	**Définition**
RFC	*Response For a Class* : nombre de méthodes invoquées en réponse à un message.
MPC	*Message Passing Coupling* : nombre de messages envoyés par une classe en direction des autres classes du système.
CBO	*Coupling Between Object* : nombre de classes avec lesquelles une classe est couplée
CBOU	*CBO Using* : se réfère aux classes utilisées par la classe cible
CBOIUB	*CBO Is Used By* : se réfère aux classes utilisant la classe cible
CBONA	*CBO No Ancestors* : CBO sans considérer les classes ancêtres.
AMMIC	*Ancestors Method–Method Import Coupling* : nombre de classes parentes avec lesquelles une classe a une interaction de type méthode-méthode et un couplage de type IC *(Import Coupling)*.
OMMIC	*Others Method–Method Import Coupling* : nombre de classes (autres que les super-classes et les sous-classes) avec lesquelles une classe a une interaction de type méthode-méthode et un couplage de type IC *(Import Coupling)*.
DMMEC	*Descendants Method–Method Export Coupling* : nombre de sous-classes avec lesquelles une classe a une interaction de type méthode-méthode et un couplage de type EC *(Export Coupling)*.
OMMEC	*Others Method–Method Export Coupling* : nombre de classes (autres que les superclasses et les sous-classes) avec lesquelles une classe a une interaction de type méthode-méthode et un couplage de type EC *(Export Coupling)*.

TAB. 1 - *Métriques sélectionnées.*

3.3 Affectation des paramètres

Pour attribuer des probabilités aux sommets, il faut distinguer deux types de variables dans le RB : les variables d'entrée et les variables intermédiaires. Les probabilités des sommets d'entrée sont déterminées directement à partir des mesures de ces variables données par un système de test. Dans notre cas, il s'agit de BOAP (Boîte à Outils pour l'Analyse de Programmes) développé au Centre de Recherche Informatique de Montréal (CRIM) (Alikacem et Snoussi, 2002). C'est un ensemble d'outils logiciels intégrés, qui permet à un expert d'évaluer rapidement le niveau de qualité d'un logiciel (faiblesses conceptuelles ou structurelles, instructions trop complexes, etc.). Le système BOAP (version 1.1.0) que nous avons utilisé contient en tout 394 classes. Les métriques considérées dans ce travail sont extraites de ce système.

Sommets d'entrée. Dans notre réseau (figure 1), les sommets d'entrée représentent ces différentes métriques. Toutes ces variables d'entrée sont des variables quantitatives qui ont des valeurs numériques mesurables. Le nombre de valeurs possibles pour ces variables peut être infini, en fonction du système de test considéré. Afin de faciliter la définition des probabilités des sommets d'entrée, nous avons besoin de transformer ces variables en variables discrètes ayant un nombre limité de valeurs (souvent « petit », « moyen » et « grand »). Plus précisément, étant donné une valeur possible d'une métrique (sommet d'entrée), le résultat de cette transformation nous donne les trois probabilités pour que cette valeur en entrée soit respectivement petite, moyenne ou grande, ce qui définit sa TPS.

Dans notre cas, nous avons utilisé la logique floue pour établir cette transformation. En effet, le processus de partitionnement flou remplace les différentes valeurs d'une métrique par un ensemble de fonctions qui représentent le degré d'appartenance (ou d'adhésion) de chaque valeur aux différentes étiquettes floues (ie, les valeurs « petit », « moyen » et « grand »). Autrement dit, le partitionnement flou généralise les méthodes de regroupement par intervalles en permettant à une valeur d'être partiellement classée dans un ou plusieurs intervalles à la fois. L'appartenance de cette valeur est distribuée dans tous les intervalles. Or dans (Trauwaert, 1988), Trauwaert a montré qu'il est possible de déterminer le nombre optimal d'intervalles avec une statistique connue sous le nom de coefficient de partition de Dunn F_k. Ce coefficient nous indique comment regrouper de la meilleure façon un ensemble de données dans différents intervalles. Plus le coefficient de Dunn est élevé, plus les sous-ensembles flous s'approchent des ensembles de la logique classique. Le nombre optimal d'intervalles (c.-à-d. de valeurs de la variable discrète en question) est donc celui qui maximise F_k. Le coefficient de partition de Dunn est calculé selon la formule :

$$F_k = \frac{1}{N}\sum_{i=1}^{N}\sum_{g=1}^{k} u_{ig}^2$$

N étant le nombre total d'observations (données), g l'indice pour un intervalle, k le nombre d'intervalles et u_{ig} la valeur ou le degré d'appartenance d'un objet donné à un intervalle.

La table 2 présente les résultats des essais de partitionnement flou avec 2 et 3 intervalles, des valeurs obtenues pour la métrique AMMIC. La partie gauche de la table 2 montre le résultat du partitionnement flou avec 2 intervalles (c.-à-d. 2 valeurs discrètes pour la métrique AMMIC : « petit » et « grand »). Pour une ligne donnée, la valeur de la première

colonne représente le numéro de l'instance (il y a en tout 394 instances ou classes) tandis que les valeurs des deuxième et troisième colonnes représentent les valeurs des degrés d'appartenance de la valeur de la métrique AMMIC aux 2 intervalles considérés, qui sont en fait les probabilités pour que cette valeur soit respectivement dans les sous-ensembles flous « Petit » et « Grand ». Il en est de même pour la partie droite de cette table sauf qu'il s'agit de trois intervalles au lieu de deux. Ces résultats montrent qu'avec deux intervalles le coefficient de Dunn est 0.82 et avec trois intervalles il est égal à 0.78. Donc, pour cette métrique, c'est le partitionnement en deux intervalles qui est retenu. D'ailleurs, c'est le même nombre d'intervalles qui a été retenu suite aux essais de partitionnement flous pour les quatre autres métriques. Pour faire le partitionnement flou, nous avons utilisé une version floue de l'algorithme *k*-means implémentée dans S-plus (Insightful, 2007).

Le degré d'appartenance à un intervalle est considéré comme la probabilité qu'une valeur de métrique appartienne à cet intervalle. En effet, (Dubois et al., 1993) montrent que la procédure bayésienne pour la mise à jour des probabilités peut-être correcte pour les observations floues. Les auteurs affirment que les degrés d'appartenance peuvent être considérés comme des probabilités avec la condition que le processus de partitionnement flou préserve la propriété qui stipule que la somme des valeurs d'appartenance aux différentes étiquettes floues soit égale à 1, ce qui est notre cas.

```
                              *** Fuzzy Partitioning ***

Membership coefficients:               Membership coefficients:
numeric matrix: 394 rows, 2 columns.   numeric matrix: 394 rows, 3 columns.
[,1]       [,2]                        [,1]        [,2]        [,3]
1 0.9873814 0.01261863                 1 0.99358023 0.004435426 0.001984347
2 0.9873814 0.01261863                 2 0.99358023 0.004435426 0.001984347
3 0.9873814 0.01261863                 3 0.99358023 0.004435426 0.001984347
…      …         …                     …      …          …           …

392 0.9873814 0.01261863               392 0.9935802 0.004435427 0.001984347
393 0.9873814 0.01261863               393 0.9935802 0.004435427 0.001984347
394 0.9873814 0.01261863               394 0.9935802 0.004435427 0.001984347

Coefficients:                          Coefficients:
dunn_coeff normalized                  dunn_coeff normalized
0.8171413  0.6342827                   0.7768965  0.6653447
```

TAB. 2 - *Exemple de résultat du partitionnement flou pour AMMIC.*

La table 3 donne un exemple de TPS du sommet AMMIC. Il s'agit d'un exemple de valeur mesurée (égale à 25) pour la métrique AMMIC. À cette valeur correspondent deux degrés d'appartenance (0.4349570 et 0.56504302) aux deux sous-ensembles flous. Ces degrés d'appartenance constituent les probabilités qui servent à définir la TPS du sommet AMMIC. Plus précisément, cette valeur d'AMMIC a une probabilité de 43% d'être dans le sous-ensemble flou « Petit », et 57% dans « Grand ».

Petit	0.43
Grand	0.57

TAB. 3 - *La TPS du sommet d'entrée AMMIC.*

Sommets intermédiaires. Les sommets intermédiaires ne sont pas directement mesurables. Ils sont définis ou influencés par leurs sommets parents. Les sommets intermédiaires ont une table de probabilité associée. Par exemple, pour un sommet *A* pouvant prendre des valeurs dans l'ensemble V^A et ayant deux parents *B* et *C* pouvant prendre des valeurs respectivement dans les ensembles V^B et V^C, chaque cellule de la table associée à *A* donne la probabilité que *A* ait la valeur $v^A_i \in V^A$ sachant que *B* a la valeur $v^B_j \in V^B$ et *C* a la valeur $v^c_k \in V^C$. Formellement

$$P\ (A = v^A{}_i \ / \ B = v^B{}_j \ \&\ C = v^c{}_k)$$

Si les tailles des ensembles V^A, V^B et V^C sont respectivement *x, y et z,* la table définit *xyz* cellules. Plus généralement, dans le cas des sommets ayant plus de deux parents, chaque cellule donne la probabilité que le sommet considéré ait une valeur sachant que chacun des parents a une valeur particulière.

Ces valeurs de probabilités peuvent être ajustées par la suite en utilisant l'apprentissage automatique à partir des données simples ou à partir des cas traités. Un parent peut influencer positivement ou négativement ses sommets fils. Les distributions des probabilités sont attribuées selon l'importance ou le poids de chacun des parents pour le sommet fils. Au départ, pour dériver les TPS il faut considérer le poids de chaque sommet parent dans la définition ou l'influence de son sommet fils. Pour cela, les TPS sont d'abord déterminées à partir des études dans le domaine et les avis des experts pour la phase initiale. Par exemple, la variable *DesignMetrics* est définie par ses deux parents *AMMIC* et *OMMIC*. Il s'agit de trouver la probabilité conditionnelle du sommet *DesignMetrics* : P (*DesignMetrics / AMMIC* & *OMMIC*). Or comme la relation entre les sommets parents *AMMIC* et *OMMIC* et leur sommet fils *DesignMetrics* est définitionnelle, la forte présence de ces métriques définit également la forte présence des métriques de conception (*DesignMetrics)*. Un scénario possible pour la TPS du sommet *DesignMetrics* est présenté à la table 4 suivante :

AMMIC	Petit		Grand	
OMMIC	Petit	Grand	Petit	Grand
Oui	0.2	0.4	0.4	0.8
Non	0.8	0.6	0.6	0.2

TAB. 4 - *La TPS du sommet intermédiaire DesignMetrics.*

Un type de raisonnement qui peut être appliqué est le suivant : si le nombre de classes parentes avec lesquelles cette classe a une interaction d'importation de type méthode-méthode est petit (*AMMIC* petit), et si le nombre de classes (autres que les superclasses et les sous-classes) avec lesquelles cette classe a une interaction d'importation de type méthode-méthode est aussi petit (*OMMIC* petit), alors la probabilité de présence de couplage de conception dans un tel système est faible ou petite. Ainsi, la probabilité de l'état « Oui » dans la table de probabilité du sommet *DesignMetrics est fixée à* 20%. Inversement, si *AMMIC* est

grand, et *OMMIC* est grand aussi, la probabilité de l'état « Oui » du sommet *DesignMetrics est fixée à* 80%. Il est important de rappeler ici qu'il y a évidemment d'autres métriques que celles considérées dans cette étude et qui sont définies soit comme métriques de conception ou d'implémentation. Ces métriques mesurent d'autres types de couplage qui peuvent influencer positivement ou négativement l'impact du changement.

4 Mise en œuvre du réseau bayésien

Une fois la structure du graphe et toutes les TPS définies, nous pouvons procéder à l'inférence bayésienne. Il en résulte une mise à jour de toutes les probabilités conditionnelles de tous les sommets. Nous avons fait appel à l'environnement BNJ (Bayesian Network tools in Java) pour atteindre cet objectif. BNJ est une suite d'outils logiciels *opensource* destinée à la recherche et au développement, en utilisant des modèles graphiques de probabilités. Elle est écrite en Java et est disponible sur le web[1].

Rappelons que notre expérimentation a été faite sur le système de test BOAP (version 1.1.0) qui contient 394 classes. Pour la mise en œuvre du réseau, nous allons choisir aléatoirement une instance à partir de laquelle nous prenons les valeurs des métriques correspondantes aux sommets d'entrée. Dès que les distributions des probabilités sont mises à jour pour les valeurs introduites, nous aurons une estimation sous forme de probabilité pour les différents états attribués au sommet *Impact* (figure 2).

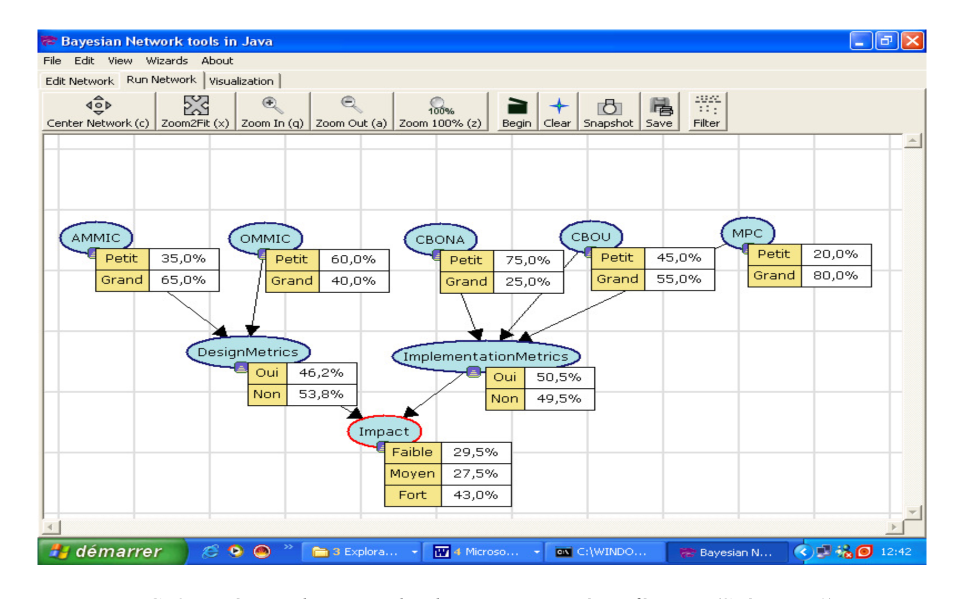


FIG. 2 - *Réseau d'Impact du changement après inférence (Scénario 1).*

[1] . http://bnj.sourceforge.net/

Ayant affecté trois états « Faible », « Moyen » et « Fort » au sommet *Impact*, et avec les données d'entrée utilisées (voir figure ci-dessus), l'exécution du réseau sur ce jeu de données a généré ces estimations d'impact (valeurs de probabilités sur le sommet *Impact*). Nous pouvons conclure par exemple que l'impact de changement a 43 % de chances d'être « Fort ».

La possibilité de faire des scénarios de la forme « qu'est-ce qui se passe si…», que les réseaux bayésiens offrent, permet d'identifier les problèmes potentiels et les actions à entreprendre.

La figure 3 montre qu'en diminuant les valeurs des métriques *CBONA* et *CBOU,* l'impact de changement s'affaiblit de plus en plus (sa probabilité d'être « faible » passe de 29,5% à 34,8%).

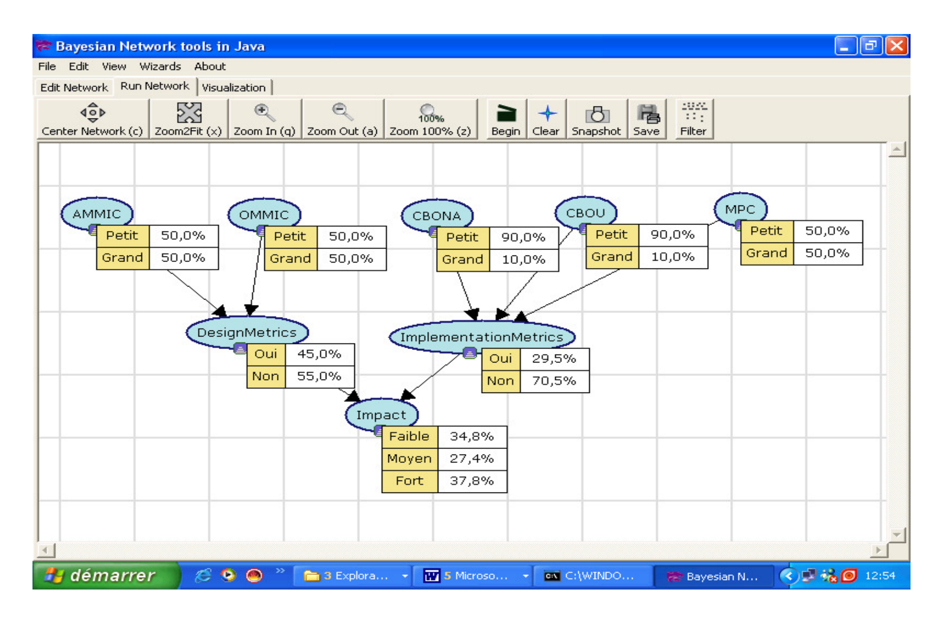


FIG. 3 - *Réseau d'Impact du changement après inférence (Scénario 2).*

La figure 4 suivante montre le scénario inverse, c'est-à-dire qu'en augmentant cette fois-ci les valeurs des métriques *CBONA* et *CBOU,* l'impact de changement devient de plus en plus fort. La probabilité de l'état « Fort » passe de 37,8% à 47,4% avec un gain de plus que 9 points (presque 10 points).

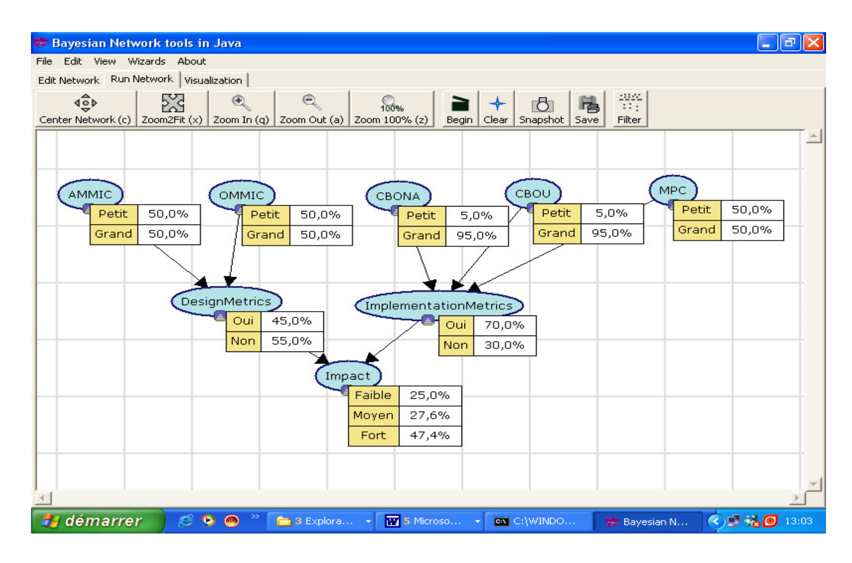


FIG. 4 - *Réseau d'Impact du changement après inférence (Scénario 3)*

Enfin, le dernier scénario exécuté (figure 5) montre qu'en maintenant petites les valeurs des métriques *CBONA* et *CBOU* et en augmentant la valeur de *AMMIC*, l'impact du changement devient un peu plus fort. La probabilité de l'état « Fort » passe de 37,8% à 41,5%.

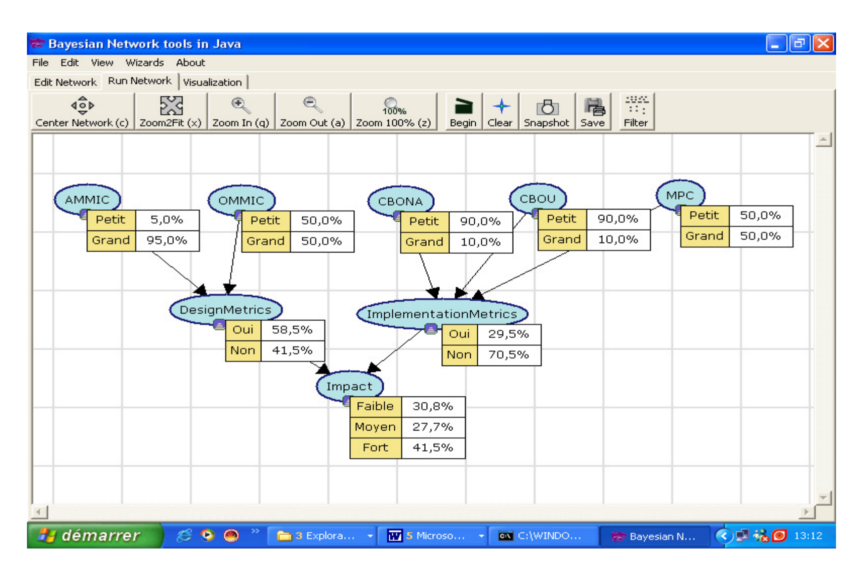


FIG. 5 - *Réseau d'Impact du changement après inférence (Scénario 4).*

Discussion. Les résultats obtenus dans les deuxième et troisième scénarios (voir figures 3 et 4) confirment bien les résultats déjà trouvés dans nos travaux antérieurs (Abdi et al., 2006a; 2006b; 2007) en utilisant une approche non probabiliste (voir respectivement règle 1 et règle 2 de la figure 6). À titre d'exemple, le résultat du scénario 3 (voir figure 4), exprimant que les métriques *CBONA* et *CBOU* influencent positivement l'impact de changement, correspond au résultat décrit par la règle 2 de causalité (Abdi et al., 2006b) comme le montre la figure 6 suivante :

Règle 1 : CBONA ≤ 3.5
CBOU ≤ 0.5
→ impact: très-faible (0.46)

Règle 2 : CBONA > 3.5
CBOU > 36.5
→ impact: fort (0.48)

Règle 3 : CBONA ≤ 3.5
CBOU ∈]0.5,1.5]
AMMIC ≤ 0.5
→ impact: faible (0.54)

Règle 4 : CBONA ≤ 3.5
CBOU ∈]0.5,1.5]
AMMIC > 0.5
→ impact: très-faible (0.76)

FIG. 6 - *Exemples de règles de causalité.*

Par contre, le résultat du scénario 4 (figure 5) ne confirme pas l'un de nos résultats (voir règles 3 et 4 de la figure ci-dessus) trouvés antérieurement dans (Abdi et al., 2006a; 2006b), car en maintenant petites les valeurs des métriques *CBONA* et *CBOU* et en augmentant la valeur de *AMMIC*, l'impact de changement ne devient pas de plus en plus faible. Sa probabilité d'être faible était à 34,8% puis elle s'est réduite à 30,8% alors qu'en principe, elle devrait augmenter. À notre avis, cela pourrait s'expliquer par le fait que l'impact du changement peut être influencé positivement ou négativement par des facteurs autres que ceux considérés dans cette présente étude. Nous pensons particulièrement à la taille du système ou sa complexité.

Parmi les contributions qui se sont intéressées à l'analyse et la prédiction de l'impact du changement, nous trouvons les travaux qui ont essayé de prédire l'impact en termes de lignes de code ajoutées/modifiées (Antoniol et al., 1999) ou de classes affectées incluant des effets de propagation (Kung et al., 1995) ou encore de changements propagés dans une classe donnée (Lee et Offutt, 1996; Lee, 1998). D'autres travaux ont essayé de donner des explications au phénomène de la propagation de modification suite à un changement effectué (Briand et al., 1999). Une troisième catégorie de travaux (Chaumun et al., 1999; Kabaili et al., 2000; Kabaili, 2002) s'est intéressée à l'étude de relations de corrélation entre d'une part, les propriétés architecturales du système (complexité, cohésion, couplage, etc.) mesurées par des métriques, et d'autre part, l'impact du changement. Des techniques statistiques ont été utilisées pour définir ces relations de corrélation.

Du point de vue des approches probabilistes, certains travaux se sont intéressés aux problématiques de la qualité du logiciel. Nous citons en particulier les travaux de Fenton et son équipe (Fenton et Neil, 2000a ; 2000b; 2001; Neil et al., 2000) qui montrent bien que les réseaux bayésiens peuvent fournir des prédictions appropriées en présence d'incertitude et de jugements subjectifs d'experts.

A travers la présente étude et nos futurs travaux, nous essayons d'évaluer ces avantages dans le cadre de l'analyse et la prédiction de l'impact du changement dans un système à objets. Nous cherchons à donner plus d'explications sur les facteurs réels et responsables de cet impact du changement ainsi que de son évolution. Nous avons déjà apporté quelques éléments de réponse à cette question dans nos travaux antérieurs (Abdi et al., 2006a; 2006b; 2007) en utilisant une approche non probabiliste. Certains résultats sont conformes aux résultats déjà trouvés, d'autres non. Enfin, nous pensons que l'exploitation des capacités d'apprentissage automatique qu'offrent les réseaux bayésiens nous permettra d'avoir dans une perspective à court terme une meilleure précision de la prédiction et ainsi des résultats plus convaincants.

5 Conclusion

Nous avons proposé dans cet article une approche probabiliste utilisant les réseaux bayésiens, afin de répondre à la problématique d'analyse et de prédiction de l'impact du changement dans un système à objets. Une étude approfondie et une synthèse générale des différents travaux antérieurs traitant de ce sujet étaient indispensables au préalable. Pour vérifier notre approche, nous avons repris une hypothèse de corrélation entre le couplage et l'impact du changement déjà vérifiée dans nos travaux antérieurs. L'expérimentation a été faite sur le système BOAP, constitué de 394 classes. Les résultats déduits de nos études empiriques (Abdi et al., 2006a; 2006b; 2007) nous ont servi pour la construction du graphe ou encore du réseau bayésien. Par la suite, nous avons défini les TPS au niveau des différents sommets d'entrée et intermédiaires. Nous avons fait appel à la logique floue pour dériver des valeurs de probabilités à partir d'un ensemble de mesures (valeurs des variables ou sommets d'entrée).

La mise en œuvre du réseau et la création de plusieurs scénarios nous ont permis de faire des prévisions sur l'impact de changement. Les résultats des deuxième et troisième scénarios ont bien confirmé des résultats déjà trouvés avec d'autres approches non probabilistes. Par contre, celui du quatrième scénario contredit l'un des résultats trouvés antérieurement (Abdi et al., 2006b). Seule une évaluation empirique sur d'autres systèmes à objets, similaires à BOAP, nous permettra d'expliquer un tel résultat. Nous envisageons d'autres expérimentations sur d'autres systèmes en considérant d'autres mesures de couplage, autres types de propriétés architecturales, ou autres facteurs qui pourraient éventuellement compléter ou mieux expliquer cette relation de causalité.

Références

Abdi, M.K., H. Lounis, et H. Sahraoui (2006a). Using Coupling Metrics for Change Impact Analysis in Object-Oriented Systems. In *Proc. QAOOSE 2006, 10th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering*, pp. 61-70.

Abdi, M.K., H. Lounis, et H. Sahraoui (2006b). Analyzing Change Impact in Object-Oriented Systems. In *Proc. 32nd EUROMICRO, Software Engineering and Advanced Applications Conference*, pp. 310-319.

Abdi, M.K., H. Lounis, H. Sahraoui, et M.K Rahmouni (2007). Vers une approche d'analyse de l'impact du changement dans un système à objets. *Revue "L'Objet",* volume 13, N° 1/2007, pp. 147-169, Éditions Hermès.

Alikacem, E. H. et H. Snoussi (2002) *BOAP 1.1.0.* Manuel d'utilisation, CRIM.

Antoniol, G., G. Canfora, et A D. Lucia (1999). Estimating the size of changes for evolving Object-Oriented Systems : a Case Study. In *Proc. 6th International Software Metrics Symposium,* pp. 250-258.

Briand, L. C., J. Wüst, et H. Lounis (1999). Using Coupling Measurement for Impact Analysis in Object-Oriented Systems. In *Proc. International Conference on Software Maintenance*, pp. 475-482.

Cantave, R. (2001). *Abstractions via un modèle générique d'application orientée objet*. Thèse de Master, Université Laval.

Chaumun, M. A., H. Kabaili, R. K. Keller, et F. Lustman (1999). A Change Impact Model for Changeability Assessment in Object-Oriented Software Systems. In *Proc. Third Euromicro Working Conference on Software Maintenance and Reengineering CSMR'99*, pp. 130-138.

Computer Society Press, Standards Collection (1993). *Software Engineering*, The Institute of Electrical and Electronics Engineers, Inc.

Dubois, D., et H. Prade (1993). Fuzzy Sets and Probability: Misunderstanding, Bridges and Gaps. In *Proc. 2nd IEEE International Conference on Fuzzy Systems,* Volume 2, pp. 1059-1068.

Fenton, N.E et M. Neil (2000a). Software Metrics: Roadmap. *In Proc. 22nd International Conference on Software Engineering*, pp. 357-370.

Fenton, N.E. et M. Neil (2000b). The Jury Observation Fallacy and the use of Bayesian Networks to present Probabilistic Legal Arguments. *Mathematics Today*, pp. 180-187.

Fenton, N.E. et M. Neil (2001). Making Decisions: Using Bayesian Nets and MCDA. *Knowledge-Based Systems,* 14, pp. 307-325.

Han, J. (1997). Supporting Impact Analysis and Change Propagation in Software Engineering Environments. In *Proc. STEP'97*, pp. 172-182.

Insightful Corporation (2007). *S-PLUS® 8 for Windows®*, User's Guide, Copyright ©.

ISO9126 (2001). ISO/IEC 9126-1 :2001, software engineering – product quality – part 1 : Quality model.

Kabaili, H., K. Rudolf, F. Lustman, et G. Saint-Denis (2000) Class Cohesion Revisited: An Empirical Study on Industrial Systems. In *Proc. QAOOSE 2000*, *4th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering*, pp. 29-38.

Kabaili, H. (2002) *Changeabilité des logiciels orientés objet : propriétés architecturales et indicateurs de qualité*. Thèse de doctorat, Université de Montréal.

Kung, D. C., J. Gao, P. Hsia, J. Lin, et Y. Toyoshima (1995). Class firewall, test order, and regression testing of object-oriented programs. *Journal of Object-Oriented Programming*, Vol. 8, No. 2, pp. 51-65.

Lee, M. et A. Offutt (1996). Algorithmic Analysis of the Impact of Changes to Object-Oriented Software. In *Proc. ICSM'96*, pp. 171-184.

Lee, M. (1998). *Change Impact Analysis for Object-Oriented Software*, PhD thesis, George Mason University.

Lindvall, M. (1999). Measurment of change: Stable and Change-Prone Constructs in a commercial C++ System. In *Proc. 6th International Software Metrics Symposium*, pp. 40-49.

Naïm, P., P. Wuillemin, P. Leray, O. Pourret, et A. Becker (2004). *Réseaux bayésiens*, Edition Eyrolles.

Neil, M., N.E. Fenton, et L. Nielsen (2000). Building large-scale Bayesian Networks. *The Knowledge Engineering Review*, 15(3), pp. 257-284.

Pfleeger, S. L. (1990). A Framework for Software Maintenance Metrics. *IEEE Transactions on Software Engineering*, pp. 320-327.

Sahraoui, H. A., R. Godin, et T. Miceli (2000). Can metrics help to bridge the gap between the improvement of OO design quality and its automation ?. In *Proc. International Conference on Software Maintenance*, pp. 154-162.

Trauwaert, E. (1988). On the meaning of Dunn's partition coefficient for fuzzy clusters. *Fuzzy Sets and Systems*, Vol.25, N° 2, pp. 217-242.

Wilde, N. et R. Huitt (1992). Maintenance support for object-oriented programs. *IEEE Transactions on Software Engineering*, Vol. 18, Issue 12, pp. 1038-1044.

Summary

In this paper, we propose a probabilistic approach using Bayesian networks to analyze and predict change impact in object-oriented systems. An impact model was built and probabilities were assigned to network nodes. Data obtained from a real system (BOAP) are exploited to empirically study causality hypotheses between some software internal attributes and change impact. With this intention, several scenarios were executed on the network. The obtained results confirmed on the one hand, some results already found in our precedent work, and on the other hand, contradicted other conclusions. Finally, we think that a probabilistic model using Bayesian nets constitutes an interesting alternative to non probabilistic models suggested in the literature.